Teaching "Professional Computing"

W. Neville Holmes
School of Computing
University of Tasmania
Launceston
neville.holmes@utas.edu.au

Abstract

Four years ago, as a result of one of a sequence of changes to the course for *Bachelor of Computing* at the University of Tasmania at Launceston, the academic staff of the School of Computing agreed to offer a first year core unit called "Professional Computing" which would give students advance notice of what will be expected of them in professional employment in the computing industry, an appreciation of the history and possible development of the computing industry, and an understanding of ethical and legal problems and issues that they would need to be aware of as professional workers in the community.

This paper reviews the content of the "Professional Computing" unit, and discusses the experience arising from three years of teaching it.

1 Background

In 1997, because of the University of Tasmania's decision to form a School of Information Systems and transfer a substantial proportion of the School of Computing's headcount to that new School, and also because of the University's fiat that course content of an "information systems" kind be removed from the School of Computing's curriculum, a substantial reduction and revision of the subject units offered by the School was inevitable. This difficult task was undertaken by the School staff during a Retreat.

The motivation behind the eventual agreement for a Professional Computing subject unit was threefold:

1. Feedback from graduates suggested that many were surprised by aspects of the computing industry they found employment in, and that this disadvantaged them in selecting jobs, and in settling into and getting ahead in their chosen profession.

2. ACS Accreditation panels had twice expressed some concern that the practice of the School in spreading its teaching of social and legal issues of computing across many of its units did not give the impression of importance the ACS felt such crucial aspects demanded.

3. Students were not well enough prepared for their large keystone project unit, which they were and are still are required to take in teams across both semesters of their final year. Although they were being given lectures on such topics as project management and teamwork as part of their project unit activity, some of this was a bit late and in any case tended to be lost in the scurry to get their projects up and going.

Consequently it was decided to offer a core first year unit called "Professional Computing" to fulfil the motivations given above. It replaced a first year core unit called "Introduction to Operating Systems" which had covered basic aspects of operating system design and gave fairly intense practical work in the use of DOS and Unix leading up to the writing of batch programs and shell scripts. The dropping of this unit was partly because of the lessening importance of DOS and partly because of the University's then new requirement that classes be in a 3 hour lecture/one hour laboratory pattern rather than the prior 2 hour/2 hour pattern.

I was given responsibility for developing and teaching the new unit, partly because I had been responsible for the dropped first year unit, and partly because of my thirty years experience in the computing industry before moving to Launceston in 1989 for the first year of

teaching of the degree, then called the Bachelor of Applied Computing.

However, the development of this unit was not at all a solo effort. I was greatly helped and encouraged at all times by my fellow staff members.

The Professional Computing unit has been delivered by me in the first semester of 1998 and 1999, and in the second semester this year, to eighty or ninety students on the Launceston campus. Around twenty students on the Burnie campus have taken the unit in my charge in 1998 and 1999, but this year they are in the charge of my colleague Paul Crowther, as also is a class of around ninety students on the Hobart campus. Paul has also had considerable experience working professionally in the computing industry.

2 Overview

The formal objectives, a hard copy of which is given to the students in their first lecture as part of their unit outline, are as follows.

> After doing this unit, a student should appreciate the context and background of professional computing, and be able in general terms to
>
> 1. discuss and explain a wide variety of uses of computers, and their social implications,
> 2. make elementary use of a variety of software representative of the broad spectrum of computer applications,
> 3. identify and describe many of the tasks and issues relevant to the profession of computing, in particular legal and ethical issues,
> 4. make presentations and reports showing understanding of some of the basic ideas and issues in computing.

The formal activities in the unit are of three kinds: *lectures* for three hours a week, *tutorials* (more in the nature of seminars) one hour a week, and intermittent *practical tests* based mostly on learning students are required to do outside the classroom. The "hours" are nominal, being actually fifty minutes. The semester is of thirteen weeks, reduced from fourteen at the time of the reform.

The lectures are designed into three streams, each of one hour per week. The final examination is based on these lectures and accounts for 70% of the overall assessment.

1. Communication and teamwork, designed to lead to skills useful in the final year project, and in employment afterwards.
2. History and ideas of computing, designed to give an appreciation of the context and scope of the computing industry, and of the ideas on which its development has been based.
3. Social and legal aspects of computing, designed to heighten awareness of the complexity of the interaction between the computing profession and the community at large.

The tutorials are designed to foster communication and presentation skills using topics about social aspects of computing. Students' work is assessed and accounts for 30% of the overall assessment.

Students are required to satisfy four practical tests to be eligible to pass the unit overall. One test involves the partial dismantling and reassembly of a personal computer, and the other three require passing very simple test sets in three quite different interactive programming languages for which they are provided with interactive tutorials on-line.

3 The Lectures

One one-hour lecture is given each week for the communications stream, and one two-hour lecture for the history/ideas stream and the social/legal aspects stream consecutively. An exception is the very first week, in which the lectures are administrative and introductory.

For the communications stream, students are required to buy a textbook [1], the excellence of which has been confirmed by numerous commendations of it by final year project students.

The communications lectures do not follow the sequence of the prescribed text. Topics beyond the text are needed. the lectures need to be given a specific computing industry slant, which the text doesn't have. And the lecture sequence is designed to fit in with the sequence of tutorial work, which work is intended to reinforce this particular lecture material.

2	presentations	8	argument
3	listening	9	reports
4	small teams	10	plain language
5	interviews	11	business
6	meetings	12	project management
7	research	13	employment

Table 1: The Communications Lectures

A second textbook [2] is required reading for the other two lecture streams though it only partly covers the material. But what it covers it covers well. This text is mainly useful for students who have come rather cold into the course and need a fairly compendious general text on computing. Of course, it is difficult for many students nowadays to be able to afford two textbooks for the one unit, and so this text has been chosen because it is also prescribed for a co-requisite first year core unit called "Computer Applications".

The history and ideas stream is sequenced and sectioned largely in the pattern of a 1974 paper on the history of data processing together with a 1997 revisit of the earlier paper.

While introducing basic ideas in the sequence of their adoption by the computing industry is only sensible, the very strongest motivation for this *time line* treatment is to counter the popular hyperbole of the computer revolution by showing the reality as evolution. Success in this is necessary to allow the students to soberly consider the future of computing, the subject of the last two lectures.

2	speech and data	8	data management
3	writing and print	9	text management
4	numbers and calculation	10	multimedia
5	unit records	11	networking
6	stored programs	12	the near future
7	operating systems	13	the far future

Table 2: The History/Ideas Lectures

One aim of the social and legal issues lectures is to couple them as far as possible to the history and ideas lectures which they immediately follow. However, the necessary prominence of a treatment of intellectual property law (facilitated by my early employment as a patent examiner), and of professional and computing ethics, makes this coupling rather weak. Early lectures on the backgound of commerce and law are more strongly so coupled.

2	oral society	8	copyright and others
3	law	9	patents
4	commerce, property	10	privacy and crime
5	contracts	11	ethics
6	identity	12	computing ethics
7	intellectual property	13	the digital industry

Table 3: The Social/Legal Lectures

4 The Tutorials

Although the first tutorial, in the second week of semester, is administrative with tutoring in the matter of the first language practical test, all the others are taken up by student presentations of one kind or another, except for tutorial classes with fewer than the normal quota of members, in which case more of the time can be used for tutoring in whatever the students request.

Each student must carry out three assignments, and these are scheduled in sequence but spread over the last eleven weeks of the semester. The first assignment focusses on an individual presentation, the second on a team-of-two presentation, and the third on a three against three formal debate.

Where a tutorial class does not provide a complete number of teams, volunteers are given the opportunity to improve their marks by taking part in an otherwise incomplete team at the end of a sequence.

4.1 Individual Assignments

Assessment value for the first assignment is divided equally between the presentation and certain work required to be submitted on a diskette.

Each student is allocated, or volunteers for, an article on which the topic is to be based, and gets one week to prepare. The articles are selected from a list provided on-line, and each student has a different article. In the list, each article is identified by an issue date, a section name, and an article title, and the student is required to find the article in the Web archives of the London Times. The articles are all related to social, legal, or other aspects of computing.

The student is required to present the topic in ten minutes using PowerPoint for visual aids. The first lectures in the communications stream give advice on such presentations. On-line tutorials are provided for PowerPoint, and in any case the co-requisite unit "Computer Applications" provides teaching in the use of this program. The student is required to present in three stages: What (the relevant substance of the article), So What (the student's view of the significance of the topic), and Now What (the student's view of the possible outcomes).

The articles, which the Times archives with their original HTML mark-up, are chosen so that most have both a graphic image included, and a link to somewhere else on the Web. Students are required to take a copy of the HTML code of the article, and any graphical inclusions, clean up the code (there is a lot of junk in such on-line materials), add material of their own, and submit their work on a diskette together with their PowerPoint file. If their article does not include a Web link or a graphical inclusion, they must get something appropriate from somewhere else.

4.2 Assignments to Pairs

Assessment value for the second assignment is also divided equally between the presentation and work submitted on a diskette. There are several differences, however.

Presentations are for fifteen minutes, and are made by pairs of students, who get two weeks to prepare. The topic is, like that of the first assignment, based on a London Times article selected from the on-line list, but

the students are required to find an issue in their article that they can present two sides of. Each team gets a different article.

The presentation is to be based on visual aids provided by PowerPoint, and their PowerPoint file is to be submitted on both their diskettes. Also on their diskettes must be their individual work diaries for the assignment, prepared as a Word document.

One purpose of this assignment is to give students experience in working with others. To make this more realistic, pairing of friends is discouraged. Another purpose is to get the students used to keeping work diaries, which they are also required to do for their final year project. For this assignment their diary is to contain their experiences and thoughts about the events and problems of their partnership.

4.3 Formal Debates

The entire assessment value for the third assignment comes from performance in the debate itself. Six students are involved in each debate, which takes up a whole tutorial. The procedures and assessment guidelines are those of formal debating in Australia.

I have been surprised that, going by a show of hands in the first Launceston lecture each year, at least a third of the students confess to some exposure to formal debating. This makes it possible for each team to include at least one member with some experience in the sport.

Each contest is based on a different social or professional issue in computing, and the teams are given two weeks to prepare.

5 The Practical Tests

In the first practical test, called *the hardware test*, pairs of students are put in charge of a personal computer in demonstrable working order. Under close supervision of technical staff, they partially dismantle the machine to the extent of removing a few circuit cards, and are required to reassemble it into working order. The purpose of this exercise is psychological—to dispel the fear and dread, or at least mystery, that people who have never looked inside a computer often feel.

All students must succeed in this test (failures can repeat, but very few need to) which is scheduled for the very first week of semester so as to avoid clashes with tutorials, which don't start till second week. Students who enrol later, or who have some strong excuse, can take the test in the last week of semester.

There are five other weeks in which practical tests are scheduled, weeks 4, 6, 8, 10, and 12. In weeks 4, 6, 8, and 10, the tests are in a different programming language each week. The students must succeed in three of the four language tests to be eligible to pass the unit. Most students will thus have completed this work by week 8. After week 10, a student who is deficient by only one language may attempt to succeed in one of the missing languages by taking tests for it in week 12.

The four languages are all interpreted, which makes them much easier to learn and test, and also made it easier to develop effective interactive tutorials which are provided for students to learn the languages from. Although there are minor demonstrations in lectures, and a small amount of assistance in tutorials, in principle students are required to learn the languages in their own time, though they are encouraged to help one another.

Students are allowed twenty minutes to finish any language test. If they do not complete the test they may repeat it if slots in later sessions are free, but they are only allowed to book into one of the three slots of a one hour test session. The tests are very simple and only take five or ten minutes for the better prepared students. The tests are published on-line from the beginning of the learning period, the only unknown factor being which of the six to ten possibilities given for each question will be marked for each student to do.

The languages are chosen for their usefulness, and for contrast in style, this last in pursuit of the variety specified in the second formal objective. The *first* language is J, used strictly in functional programming mode (J Software calls this *tacit programming*) as required in an Honours unit treating functional computation. The *second* is the logic programming language Prolog, used in a second year artificial intelligence unit. The *third* is the database language SQL, which many students will have to use in professional employment, and which some need in their final year project. The *fourth* is the object-oriented system programming language Python, which seems to be gaining wide acceptance and is preferred by many to the similar but better-known Perl.

Students are only required to learn a mere smidgen of each language. Only lists of numbers are manipulated in their J work, and only a fraction of the available functions and operators is needed. Students are provided with a file of ready-to-consult assertions for their Prolog work, and need only consult and make simple assertions. For SQL a set of tables (based on Georgette Heyer's Regency novels) is provided, and the student need only learn simple select statements. For Python, only very simple objects are used.

6 Some Observations

The following observations are segregated according to the activity being commented on. Otherwise they are sequenced as they came to mind during the writing of

the paper. Thus there is no implication of relative importance to be drawn from the sequence of observations.

6.1 The Lectures

The lectures were delivered on two campuses in 1998 and 1999, and on three in 2000. In 1998 and 1999 videoconferencing was used to deliver lectures to the Burnie campus, where there were only fifteen or twenty students. The difficulties of this technique, which the University is pressing lecturers to use, for the relatively discursive subject matter of this unit was such that I felt it necessary to travel to Burnie for occasional weeks of face to face lecturing to maintain rapport with the students.

The experiences with videoconference lecturing to a small campus for this particular unit led me to insist that the lectures to the large Hobart campus class in 2000 be all face to face. For quite other reasons the Burnie lectures for 2000 were also all face to face.

At present the lectures are given almost entirely with visual aids of the *bullet point* kind. They would undoubtedly be improved by the addition of illustrations. This would require a careful redesign of the lectures, which would be impossibly time-consuming under present circumstances.

6.2 The Supplementary Material

However, my perception of the need for material to amplify and illustrate the lecture material, and of the difficulty the overseas students felt in facing up to a wide spectrum of subject matter in a foreign language, led me to start developing on-line supplementary material. Another motivation for providing extra study material was the great increase of students in the year 2000. Development of such material is also very time-consuming, but has the advantage that it can be put together incrementally and still be used by students as it is being developed.

Development of this supplementary material was started early this year 2000 but is still very fragmentary. At the time of writing there has been little feedback from students which would either encourage or discourage or modify this approach to providing learning materials.

What has been done is as follows. There is a base HTML file comprising a table of lectures much as is given above. Each entry in the table links into an HTML file for that lecture week. The main bullet points for each lecture are given as headings. The idea is that the headings, and any point given under them, may be linked to Web resources, to private resources, or to relevant points in one of the two HTML index files.

The two index files, one for technical terms and one for proper names, are arranged like dictionaries or glossaries, with the links for each entry pointing, as above, to Web resources or to private resources. There is also some cross-linking. The private material is a collection of papers and articles drawn from a variety of sources, and this material, like all the supplementary material, is only accessible to our own students.

6.3 The Tutorials

The archives of the London Times are solely used as a source of assignment articles because having one source is simpler for students, because the articles are well suited to the requirements of the assignments, and because the archives remain entirely open to the public.

For the first and second assignments students must submit certain work on a diskette. By this means, all students get experience with use of diskettes, which they wouldn't get if they were allowed to submit their work by e-mail, as many would rather do.

Students are expected to learn to understand HTML mark-up. They are pointed to tutorials on the subject, and they are required to edit their own HTML files to remove junk code.

In 1998 and 1999 the articles for the second assignment were pulled from various places, such as computing journals, and put on-line for students to use. However, the extra students in 2000 has forced this assignment to be based on the Times archives, from which suitable (though lighter) articles are easier to glean.

In 1998 and 1999 the pairs were given one week to prepare for asignment two. The idea was that the short notice would allow their efforts to be concentrated. There were, of course, problem pairs and some of these pleaded shortage of time. Therefore the time was extended to two weeks for assignment two in 2000 to allow pairs more time to get their act together. This will not eliminate the problem pairs altogether.

The difficulties of getting students to work together, particularly in teams of three, are many, and the tutors in charge of these activities must be careful to maintain control of the various situations, and to be quick to detect problems and fix them. Fortunately, most of the students find these activities enjoyable, particularly the debates, and so most of them are very cooperative.

One of the greatest difficulties is the scare factor for bashful students and for students with no prior experience of speaking in front of an audience. Some students have tried vigourously to avoid speaking at all. This is why the rule is made for the first two assignments that if there is no presentation, no marks at all are awarded. To compensate, there is also a rule that if the student

actually presents for roughly the full time, no matter how badly, a pass mark is awarded for the presentation.

The students with extreme degrees of scaredness can usually be persuaded on the grounds that they will certainly be required to make presentations in their professional employment, and they are better off giving their first and most scary presentations before a relatively familiar and friendly audience rather than postponing it until they have to present to strangers. In only one case have we had to allow a student to make his first presentation to an audience of merely the lecturer and the tutor.

6.4 The Practical Tests

The hardware test is an unqualified success. So greatly was it appreciated that in 1998 we were petitioned by second and third year students to run sessions for them, and the extra sessions were well attended.

The language tests have been difficult in several ways. The main difficulty is that many students, particularly but not only the foreign ones, are very uncomfortable with, if not anguished by, the idea of having to learn without being specifically taught. Attempts are made in lectures to counter this attitude by telling them that there are hundreds of different programming languages in use in the industry with more being adopted year in year out, and that in professional employment they will in all likelihood have to deal with completely unfamiliar programming languages from time to time without the benefit of having interactive tutorials to help them learn. They are told that the language tests are intended to prepare them for this reality.

Another difficulty is that some students seem very reluctant to seek help from their tutor or lecturer. Students failing a test will often, when asked, confess that they really hadn't understood what they were trying to do, but that they had been too afraid to ask help. They claim ignorance of the regular consultation times set aside by lecturers and announced in lectures, times which are quite fruitful for the few students who do seek help. Some students try to learn all the possible answers by heart, which is much more difficult than learning how the language works, and rumour has it that there is a good deal of cheating at which one or two have been caught.

Despite this, it must be said that these are problems besetting the minority. Many students express delight in the experience, and some final year students have described the experience as particularly valuable, seen in retrospect.

For all languages except SQL students use early public domain versions of the interpreters and thus may load the interpreters and tutorials onto their own machines. The SQL used is the one hidden within Microsoft Access, and so it is, unfortunately, much more difficult and ponderous to use than the others. The tutorials for SQL are constructed as SQL tables, however, so students who have Access on their own machines may use them there. Interestingly, several such students have remarked that they had never suspected that such an excellent capability was available to them there.

I have started having doubts about the sequencing of languages. Using J first was in the belief that it was the simplest to learn, as indeed it seems to be for many students. But it is frankly computational in the arithmetic sense, and it is this aspect that many students seem to prop at. It would seem that the school system can no longer be relied on to impart numeracy to the bulk of students, and, in deference to local conditions, the Bachelor of Computing course has never required any particular study of mathematics for entrance to the course. It would seem better to place J third, so that relatively innumerate students would not be simultaneously challenged by the new idea of learning rather than being taught, and by having to deal arithmetically with numbers.

In 1998 and 1999 several test sessions were scheduled for all weeks from 3 to 12 inclusive. Despite continued urging, very very few students attended tests in the odd numbered weeks, so these were dropped altogether in the year 2000.

7 Conclusion

Experience with teaching "Professional Computing" has been both demanding and rewarding.

It is demanding for the lecturer simply because the breadth of material presented means that rehearsal and review for the lectures is time consuming every year. One is also frequently coming across interesting material that deserves inclusion somewhere in the lectures. Trying to exploit and keep control of such material can be very challenging.

Not only the breadth of the material, but also the indefiniteness of much of it, is particularly daunting for computing students, and it is difficult for the lecturer to ameliorate this. Two countermeasures have seemed at least partly effective.

The *first* countermeasure is, from the beginning, to stress repeatedly to the students in lectures that they are being prepared for professional employment, that any of them unfortunate enough to be employed merely to write programs would in effect be plying a trade, and that professional people are employed principally to advise other people are valued for their opinions and

judgement, not for their factual knowledge which can equally well be acquired by journalists and hobbyists, and even by fools.

The overarching purpose of the "Professional Computing" unit is thus, the students must be emphatically told, to ready them to form reasonable opinions and judgements and to be able to present those opinions and judgements to clients and the public alike. Such is the nature of a profession. Students must be told, therefore, that they will be given opportunities in the final examination to express opinions, which opinions will be judged on their quality rather than on their specific content.

A great difficulty is that some students, almost invariably ones fresh from school, see these teachings as irrelevant. Many of them stay away from lectures on that basis, so there is no chance to motivate them directly. They are often quite bright otherwise, and are able to pass the practical tests and complete their assignments with a minimum of effort. Such students, as far as one can tell, make up a high proportion of those who fail the unit because they do very poorly in the final examination. This is very sad, and was made even sadder for me when I was harangued by such a student for setting an unprofessional examination—it should, he vehemently asserted, have been made up of questions on useful topics such as Microsoft Windows.

The *second* countermeasure is to urge students repeatedly in lectures to study the past and sample examination papers provided on-line so that they can judge from the questions what is expected of them, and to assure them that they would see that there is no need to struggle to remember all the material gone over in the lectures. Rather they should be thinking about what they are being told to form their own opinions of it, to discuss issues in computing with fellow students, and to take advantage of their tutorials to discuss issues in class.

It has proved also to be rewarding to teach "Professional Computing", not just from any sense of doing the right thing for students whether they like it or not, but from observing the enjoyment that the majority of them get from being challenged and from increasing their self-confidence.

This year, for the first time, we have final year students who have done "Professional Computing". The feedback I have got so far from them has been very pleasing, and I look forward to getting feedback from graduates in professional employment.

References

[1] Cielins, M. and Aquino, M. *The Business of Communicating.* Irwin/McGraw-Hill, Sydney, fourth edition (1999).

[2] Parsons, J.J., Oja, D., and Low, S. *Computer Technology and Society*, Course Technology, Cambridge MA, second edition (1999)

In the interest of maintaining my anonymity, and because I have not had, nor will have, the time to do the thorough and other-centred bibliography I would prefer, this reference list and the citations that refer to them are restricted to the two textbooks. A much longer, though regrettably egocentric, reference list will be provided in the final version of the paper, should it be accepted for ACE2000.

I also intend to solicit comments from my supporting lecturer for inclusion in the final version of the paper. As he is at the moment only a few weeks into lecturing in the unit, it is too soon to ask for comments now.